

**Universitatea Politehnica Bucuresti, Facultatea de Automatica si  
Calculatoare**

# **Proiectare arhitecturala – Talk2Me**



**Disciplina: Ingineria Programelor**

**Membrii echipei:**

1. **Dumitrescu Evelina** 331CA evelina\_dumitrescu@yahoo.com
2. **Murgoci Gabriela** 334CA gabriella.murgoci@gmail.com
3. **Sandulescu Dragos** 331CC sandulescu\_dragos@yahoo.com
4. **Stingu Lavinia Andreea** 334CA andreeastingu@yahoo.com

# Cuprins

## 1. Tehnologiile folosite

1.1	Limbajul de programare.....	3
1.2	Medii de dezvoltare.....	3
1.2.1	Windows Presentation Foundation.....	3
1.2.2	XAML .....	3
1.2.3	Microsoft Expression Blend .....	4
1.2.4	Socketi TCP.....	4
1.2.5	Platforma Azure – SQL Azure.....	4

## 2. Module de nivel inalt

2.1	ChatWindow.....	6
2.2	LoginWindow.....	6
2.3	DataBaseManager.....	7
2.4	PrintFile.....	8
2.5	EmailSender.....	9
2.6	User.....	9
2.7	Admin.....	12
2.8	Password Recovery Window.....	12
2.9	Admin Window.....	13
2.10	Client Window.....	14
2.11	ConversationHistory.....	14
2.12	Login.....	15
2.13	Authentication.....	15
2.14	Message.....	16
2.15	FriendOp.....	16
2.16	StateStatus.....	17
2.17	ClientServerCommunicator.....	17

2.18	Server.....	18
<b>3. Use Case</b>		
3.1	Autenticare.....	20
3.2	Chat.....	21
3.3	Administrare.....	21

## 1. Tehnologiile folosite

### 1.1 Limbajul de programare

Proiectul Talk2me este realizat in limbajul de programare C#.

**C#** este un limbaj de programare orientat-obiect conceput de Microsoft la sfarsitul anilor 90. A fost conceput ca un concurent pentru limbajul Java. Ca si acesta, C# este un derivat al limbajului de programare C++.

Am ales sa dezvoltam aplicatia in C# datorita usurintei pe care o ofera in programare prin pachetele implicite, documentatiei bine scrise, modului facil de a crea o interfata grafica.

### 1.2 Medii de dezvoltare

#### 1.2.1 Windows Presentation Foundation

WPF sau *Windows Presentation Foundation* este un API utilizat la construirea interfetelor grafice utilizator (UI) pentru aplicatiile desktop cu ajutorul a .NET Framework.

WPF combina elemente din web development, de exemplu limbajul de marcare pentru interfetele utilizator declarative, cu elemente din Rich Internet Applications, cum ar fi grafica vectoriala scalabila, animatia si suportul media.

Principalele caracteristici ale WPF :

- Interfata utilizator declarativa (Declarative UI) – construirea interfetei se realizeaza utilizand limbajul de marcare XAML.
- Pozitionare inteligenta – WPF ofera un sistem de paginare extensibil pentru a aranja vizual elementele unei interfete utilizator.
- Grafica scalabila – este bazata pe vectori.
- Template-uri – exista doua tipuri de template-uri in WPF : *control template* si *data template*.
- Legarea (Binding) – data binding, dar de asemenea permite legarea declarativa a comenzilor, animatiilor si evenimentelor.
- Styling – stiluri bogate, aplicatiile pot arata uimitor.
- Declansatoare (Triggers) – permit dirijarea schimbarilor de stare; utile pentru animatii.
- Animatii – majoritatea proprietatilor in WPF pot fi animate, exista suport pentru timeline, key frames si interpolare. Animatiile se integreaza usor cu stilurile si template-urile.
- 3D – permite unele modelari 3D de baza.

#### 1.2.2 XAML

Limbajul XAML, sau Extensible Application Markup Language, este un limbaj dezvoltat de Microsoft. Acest limbaj este bazat pe XML si este fundamental pentru WPF. Acronimul initial venea de la Extensible Avalon Markup Language, Avalon fiind numele de cod pentru Windows Presentation Foundation.

XAML (se pronunta *zammel*) este utilizat pentru crearea interfetelor utilizator in WPF. XAML este utilizat de asemenea de Windows Workflow Foundation (WF) pentru definirea de workflow-uri. In sensul cel mai general, XAML este un limbaj pentru serializarea obiectelor .NET intr-un format lizibil oamenilor. Extensia de fisier implicit este *.xaml*.

XAML poate fi in functie de modul in care este utilizat, fie compilat, fie interpretat.

### 1.2.3 Microsoft Expression Blend

Al doilea instrument major pentru crearea aplicatiilor WPF de la Microsoft este Expression Blend. Blend este oarecum comparabil cu IDE-ul pentru Adobe Flash. Vom gasi aici instrumente de desenare, linii de timp pentru animatii, palete si alte functii centrale. Blend este, de asemenea, unul din primele produse de la Microsoft scrise cu WPF.

### 1.2.4 Socketi TCP

Socketii sunt mijloace generalizate de comunicatie interproces deoarece: procesele comunicante pot rula pe alte masini, arhitectura masinii poate fi diferita, sistemul de operare poate fi diferit. Canalul permite comunicatia bidirectionala intre procese, dar nu se ocupa de structura datelor transmise. Un socket non-blocant este un socket ce trimite un apel si executia programului va continua.

In general, intr-un sistem bazat pe comunicatia cu socketi serverul urmeaza anumiti pasi pentru initierea comunicatiei:

- Creeaza un socket
- Asociaza socketului o adresa
- Ascuta cererile clientilor

Cand un client doreste sa initieze o comunicatie cu serverul, acesta urmeaza si el anumiti pasi:

- Creeaza un socket
- Determina locatia serverului (adresa si port)
- Incepe trimiterea si/sau primirea datelor

### 1.2.5 Platforma Azure – SQL Azure

Windows Azure este o platforma de cloud computing si infrastruktura, creata de Microsoft, pentru construirea, implementarea si gestionarea aplicatiilor si serviciilor prin intermediul unei retele globale de centre de date gestionate de Microsoft. Windows Azure poate fi folosit pentru a construi o aplicatie web care ruleaza si stocheaza datele sale in centrele de date Microsoft.

Platforma Windows Azure este alcatuita din:

- Windows Azure: un sistem de operare, ca serviciu
- SQL Azure: o baza de date relationala completa in Cloud
- AppFabric: servicii de tip Web orientate spre consumator care ofera conectivitate sigura si control al accesului federalizat pentru aplicatii.

Windows SQL Azure este un serviciu cloud-based de la Microsoft ce ofera capacitati de stocare a datelor ca parte a Azure Services Platform. Spre deosebire de baze de date bazate pe cloud similare, SQL Azure permite utilizatorilor sa faca cereri relationale cu datele stocate, care poate fi structurat sau semi-structurate, sau chiar documente nestructurate. SQL Azure ofera interogarea datelor, cautare, analiza de date si sincronizare de date.

## 2. Module de nivel inalt

### 2.1 ChatWindow

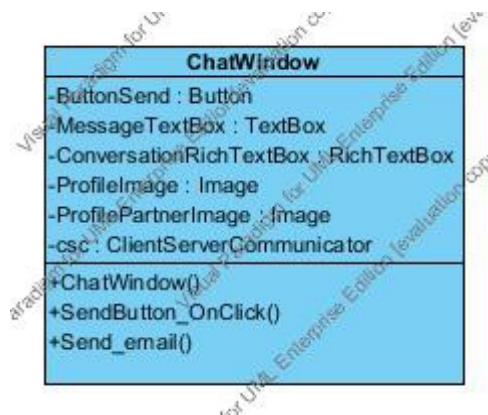
Clasa mosteneste un obiect de tip Window si reprezinta fereastra prin intermediul careia utilizatorul realizeaza schimbul de mesaje cu alti utilizatori.

Membrii clasei sunt de tip:

- Button ButtonSend pentru trimiterea mesajului scris in MessageTextBox
- TextBox MessageTextBox este case pentru scrierea mesajului
- RichTextBox ConversationRichTextBox pentru afisarea conversatiei
- Image ProfileImage pentru afisarea pozei de profil
- Image ProfilePartnerImage pentru afisarea pozei de profil a partenerului de conversatie
- ClientServerCommunicator pentru comunicarea cu serverul

Metodele clasei:

- SendButton\_OnClick() se trimite textul din MessageTextBox
- Send\_email() pentru trimitere email catre contactul selectat



## 2.2 LoginWindow

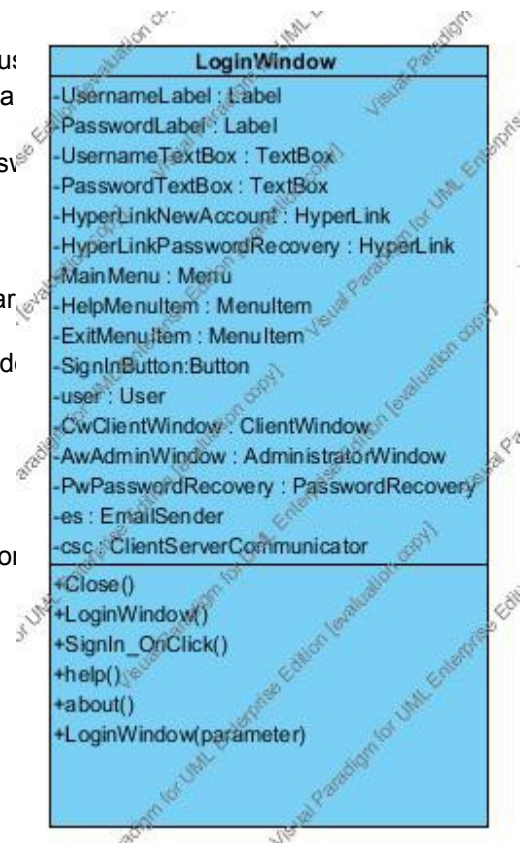
Clasa mosteneste un obiect de tip Window, fiind o fereastră de sine statatoare.

Membrii clasei sunt:

1. controale de tip:
  - TextBox si Label pentru completarea credentialelor (u
  - CheckBox pentru validarea optiunilor de memorare a automatic sign in
  - HyperLink pentru crearea unui nou cont si pentru pass
  - MainMenu pentru optiuni de Help si Exit
  - Button pentru autentificare
2. un obiect de tip ClientServerCommunicator pentru realiz
3. obiect de tip User pentru obtinerea informatiilor din baza d

Metode:

- constructor LoginWindow()
  - initializarea obiectului de tip ClientServerCommunicator
- Close()



- pentru inchiderea ferestrei
- SignIn\_OnClick()
  - se apeleaza atunci cand apare un eveniment OnClick pentru butonul de sign in
  - apeleaza ClientServerCommunicator pentru validarea autentificarii
  - in caz de succes se obtin datele referitoare la utilizator si se deschide o noua fereasta in functie de tipul de utilizator
  - in caz de esec, se afiseaza un MessageBox cu mesajul de eroare si cere utilizatorului reintroducerea datelor
  - in caz de reusita
- help()
  - afisare fereasta de Help
- about()
  - afisare AboutBox

## 2.3 DataBaseManager

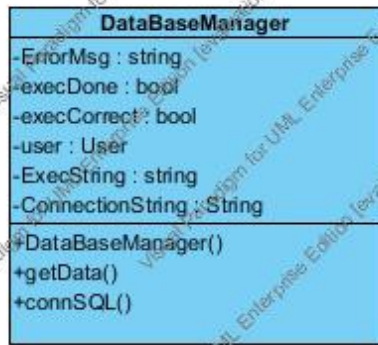
Este o clasa generica pentru manipularea bazei de date.

Membrii clasei:

- ErrorMessage
  - pentru intoarcerea mesajului de eroare in cazul in care una din operatii esueaza
- execDone
  - este setat pe true atunci cand se termina executia
- execCorrect
  - este setat pe true daca toate comenzile au fost executate cu succes
- user
  - in urma executiei se va returna un obiect de tip User sau null
- ExecString
  - sql ce trebuie executata
- ConnectionString
  - string pentru realizarea conexiunii la baza de date

Metode:

- DataBaseManager()
- getData()
  - executa comanda SQL si intoarece un obiect de tip User
- connSQL()
  - realizeaza conexiunea la baza de date



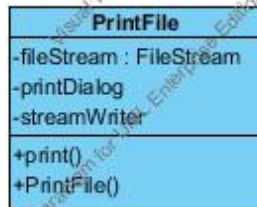
## 2.4 PrintFile

Membrii:

- `fileStream`  
- obiect `FileStream`
- `printDialog`  
- obiect de tip `PrintDialog` pentru afisare fereastra cu optiuni de printare
- `streamWriter`  
- obiect de tip `StreamWriter` pentru scrierea in fisierul `fileStream`

Metode:

- `print()`  
- printeaza fisierul cu istoricul conversatiilor selectat



## 2.5 EmailSender

Metode:

- `sendMail()`  
- creeaza un nou obiect de tip `System.Net.Mail.MailMessage`, completeaza corespunzator headerul si continutul emailului si il expediază catre destinatar





## 2.6 User

Clasa User stocheaza toate campurile din baza de date corespunzatoare unui utilizator (reprezinta interfata de comuniace dintre baza de date si aplicatie).

Membrii:

- String username  
- stocheaza numele
- String password  
- stocheaza parola
- String gender  
- stocheaza sexul
- String birthdate  
- stocheaza ziua de nastere
- String birthplace  
- stocheaza locul de nastere
- String telephone  
- stocheaza numarul de telefon
- String personal\_interest  
- stocheaza hobby-urile
- String education  
- stocheaza informatiile legate de studii
- String workplace  
- stocheaza locul de munca
- String current\_city  
- stocheaza domiciliul curent
- String country  
- stocheaza tara natala
- String address  
- stocheaza adresa curenta
- String nationality  
- stocheaza nationalitatea

- String languages  
- stocheaza limbile cunoscute/vorbite
- String groups\_friends  
- stocheaza lista de grupuri si prieteni
- String secret\_question  
- stocheaza intrebarea secreta necesara recuperarii parolei
- String secret\_answer  
- stocheaza raspunsul la intrebarea secreta necesara recuperarii parolei

Metode:

1. care seteaza variabilele mai sus mentionate:

- Set\_gender
- Set\_birthdate
- Set\_birthplace
- Set\_telephone
- Set\_personal\_interesat
- Set\_education
- Set\_workplace
- Set\_current\_city
- Set\_country
- Set\_address
- Set\_nationality
- Set\_languages
- Set\_favourite\_quotes
- Set\_groups\_friends
- Set\_secret\_question
- Set\_secret\_answer

2. care returneaza valorile mentionate:

- Get\_gender
- Get\_birthdate
- Get\_birthplace
- Get\_telephone
- Get\_personal\_interesat
- Get\_education
- Get\_workplace
- Get\_current\_city
- Get\_country
- Get\_address
- Get\_nationality
- Get\_languages
- Get\_favourite\_quotes
- Get\_groups\_friends
- Get\_secret\_question

- Get\_secret\_answer



## 2.7 Admin

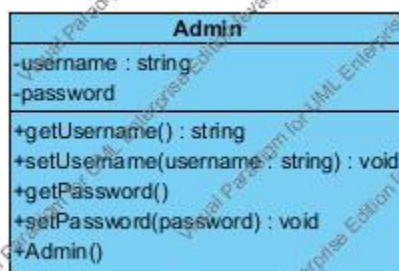
Este clasa ce reprezinta un user administrator.

Membri clasei:

- username si password de tip string reprezinta credentialele necesare autentificarii

Metodele clasei:

- pentru variabilele mai sus r



## 2.8 PasswordRecoveryWindow

Clasa mosteneste un obiect de tip Window.

Membrii clasei sunt:

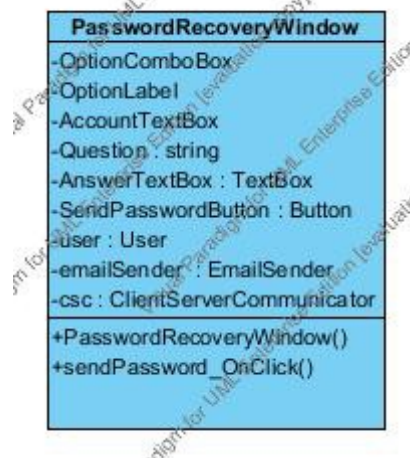
1. controale de tip:

- ComboBox OptionComboBox pentru selectarea alternativa a username-ului sau a adresei de e-mail necesare pentru identificarea contului
- Label OptionLabel pentru afisarea optiunii selectate de ComboBox
- TextBox AccountTextBox pentru username sau adresa de mail
- Label Question pentru afisarea intrebării secrete
- Textbox TextBoxAnswer pentru afisarea raspunsului la intrebarea secreta
- Button SendPasswordButton pentru trimitere cerere de recuperare parola

1. ClientServerCommunicator csc pentru realizarea comunicatiei cu baza de date prin intermediul serverului
2. User user pentru stocarea raspunsului interogarii bazei de date
3. EmailSender emailSender pentru posta electronica

Metode:

- SendPassword\_Click()
- interogheaza baza de date si trimite e-mailul cu parola utilizatorului in cazul in care informatiile furnizate sunt valide



## 2.9 Admin Window

AdminWindow extinde un obiect de tip Window si este destinat administrarii mesajelor de log si al utilizatorilor.

Membrii clasei sunt:

- TabControl AdminTabControl ce contine 2 obiecte de tip TabItem, LogTabItem si RemoveUserTabItem
- LogTabItem contine un obiect de tip ListBox LogListBox in care vor fi listate logurile si un Button ListLogsButton
- RemoveUserTabItem contine un TextBox UserDataTextBox pentru selectarea unui user, un Button ApplyButton pentru confirmarea operatiei de stergere si un Button pentru selectarea criteriilor de stergere
- ClientServerCommunicator pentru realizarea comunicatiei cu baza de date cu ajutorul serverului

Metodele clasei sunt:

- ListLogsButton\_OnClick() pentru actualizarea fisierelor log
- ApplyButton\_OnClick() pentru validarea operatiei de stergere



## 2.10 Client Window

ClientWindow extinde clasa Window si este fereastra ce apare in urma operatiei de autentificare.

Membrii clasei sunt:

- ListBox ContactListBox pentru afisarea listei de contacte si a grupurilor
- TextBox SearchTextBox pentru cautarea interactiva a contactelor
- Menu MainMenu pentru schimbarea preferintelor legate de profil , help si exit
- obiect de tip ConversationHistory pentru vizualizarea istoricului conversatiilor

Metodele clasei:

- ContactListBox\_OnClick() pentru selectarea contactului cu care va avea loc schimbul de mesaje
- MainMenu\_OnClick() pentru deschiderea ferestrei corespunzatoare setarii profilului
- OnExit() inchiderea aplicatiei
- Help() deschidere fereastra pentru asistenta tehnica



## 2.11 ConversationHistory

Clasa extinde un obiect de tip Window si gestioneaza istoricul conversatiilor pentru fiecare user in parte.

Membrii clasei sunt:

1. controale de tip:

- ListBox : ContactListBox pentru afisarea contactelor cu care a avut loc schimbul de mesaje si DateTimeListBox pentru afisarea datei si orei la care au avut loc
- RichTextTextBox ConversationRichTextBox pentru afisarea conversatiei
- MainMenu de tip MenuItem pentru stergerea si printarea conversatiilor

2. obiect de tip PrintDocument pentru printarea fisierului ce contine conversatia

Metodele clasei :

- ContactListBox\_OnClick pentru selectarea contactului cu care a avut loc conversatia
- DateTimeListBox\_OnClick pentru selectarea datei, orei si adresei de contact cu care a avut loc conversatia.



## 2.12 Login

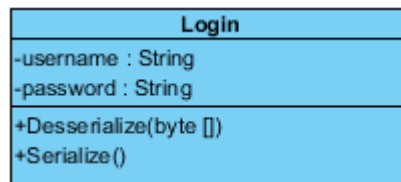
Contine datele necesare pentru a trimite un mesaj de login.

Membrii clasei sunt:

- doua Stringuri ce contin username si parola; acestia vor fi instantiati in constructorul clasei cu valorile primite ca parametru.
  - String username
  - String password

Metodele clasei :

- public static LoginDesserialize(byte[] data) ;
- public byte[] Serialize();



## 2.13 Authentication

Contine datele necesare pentru a trimite un mesaj de autentificare.

Membrii clasei sunt:

- trei stringuri ce contin id-ul si parola si adresa de e-mail pentru recuperarea parolei; acestia vor fi instantiati in constructorul clasei cu valorile primite ca parametru.
  - String username
  - String password
  - String email

Metodele clasei :

- public static Login Desserialize(byte[] data) ;
- public byte[] Serialize();

Authentication
-username : String
-password : String
-email : String
+Desserialize(byte [])
+Serialize()

## 2.14 Message

Contine datele necesare pentru a trimite un mesaj de autentificare.

Membrii clasei sunt:

- trei stringuri ce contin id-ul clientului destinatie, sursa si altul cu mesajul; acestia vor fi instantiati in constructorul clasei cu valorile primite ca parametru.
  - String message
  - String source\_name
  - String destination\_name

Metodele clasei:

- public static Login Desserialize(byte[] data) ;
- public byte[] Serialize();

Message
-message : String
-source_name : String
-destination_name : String
+Desserialize(byte [])
+Serialize()

## 2.15 FriendOp

Contine datele necesare pentru a realiza adaugarea, stergerea, mutarea unui prieten in alt grup.

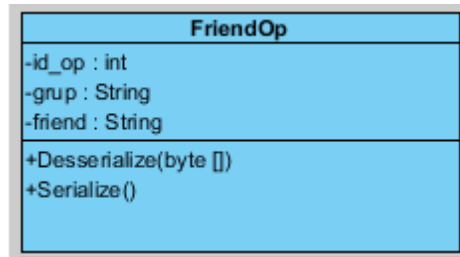
Membrii clasei sunt:

- un int ce contine id-ul operatiei, un string cu grupul si un string cu prietenul ; acestia vor fi instantiati in constructorul clasei cu valorile primite ca parametru.
  - Int id\_op
  - String grup
  - String friend

Metodele clasei:

- public static Login Desserialize(byte[] data) ;
- public byte[] Serialize();





## 2.16 StateStatus

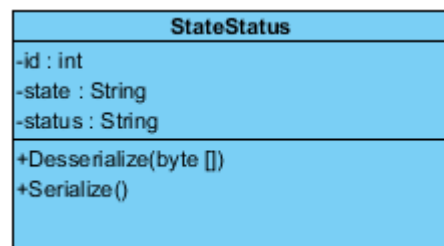
Contine datele necesare pentru a seta starea sau statusul.

Membrii clasei sunt:

- un int ce contine id-ul operatiei, un string cu starea sau statusul ; acestia vor fi instantiati in constructorul clasei cu valorile primite ca parametru.
  - Int id
  - String state
  - String status

Metodele clasei:

- public static Login Desserialize(byte[] data) ;
- public byte[] Serialize();



## 2.17 ClientServerCommunicator

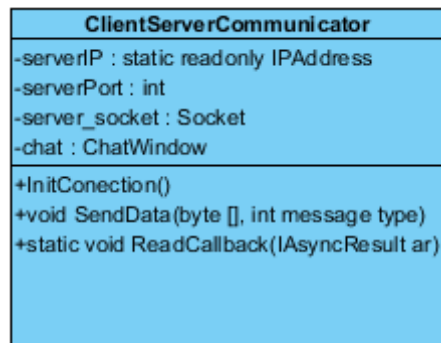
Realizeaza conexiunea la un server si trimite si accepta mesaje.

Membrii clasei sunt:

- static read-only IPAddressserverIP
  - adresa IP
- const int serverPort
  - port server
- Socket server\_socket
  - obiect de tip socket pentru realizarea conexiunii
- ChatWindowchat;
  - in cazul primirii unui mesaj text in RichTextBox se va adauga informatia utila

Metodele clasei:

- void InitConection()  
-Initializare conexiunii si crearea obiectului de tip socket
- void SendData(byte[] message, int message type)  
-trimiterea datelor se face dupa un anumit format; prima parte a mesajului va contine id-ul mesajului si dimensiunea acestuia, iar restul vor fi doar date.
- static void ReadCallback(IAsyncResult ar)  
-fiecare mesaj va contine in primele doua campuri dimensiunea mesajului si tipul mesajul, iar aceasta functie va astepta pana cand mesajul a fost primit complet si apoi va fi deserializat. Pentru mesajele tip text noua informative va fi salvata in campul RichTextBox din cadrul ferestrei.



## 2.18 Server

Acest modul realizeaza functionalitatile unui server care va asculta pe un anumit port. Acesta va prelua mesajele si le va salva in cazul in care aceste nu se pot trimite clientului destinatie.

Membrii clasei:

- const int server\_Port  
- port server
- Socket server\_socket  
- obiect de tip socket pentru realizarea conexiunii
- DataBaseManager database  
- pentru operatiile cu baza de date
- Dictionary<name, Socket> map  
-pentru a identifica socket-ul pentru un anumit client destinatie in cazul in care se trimit mesaje

Metodele clasei:

- void BindConection()

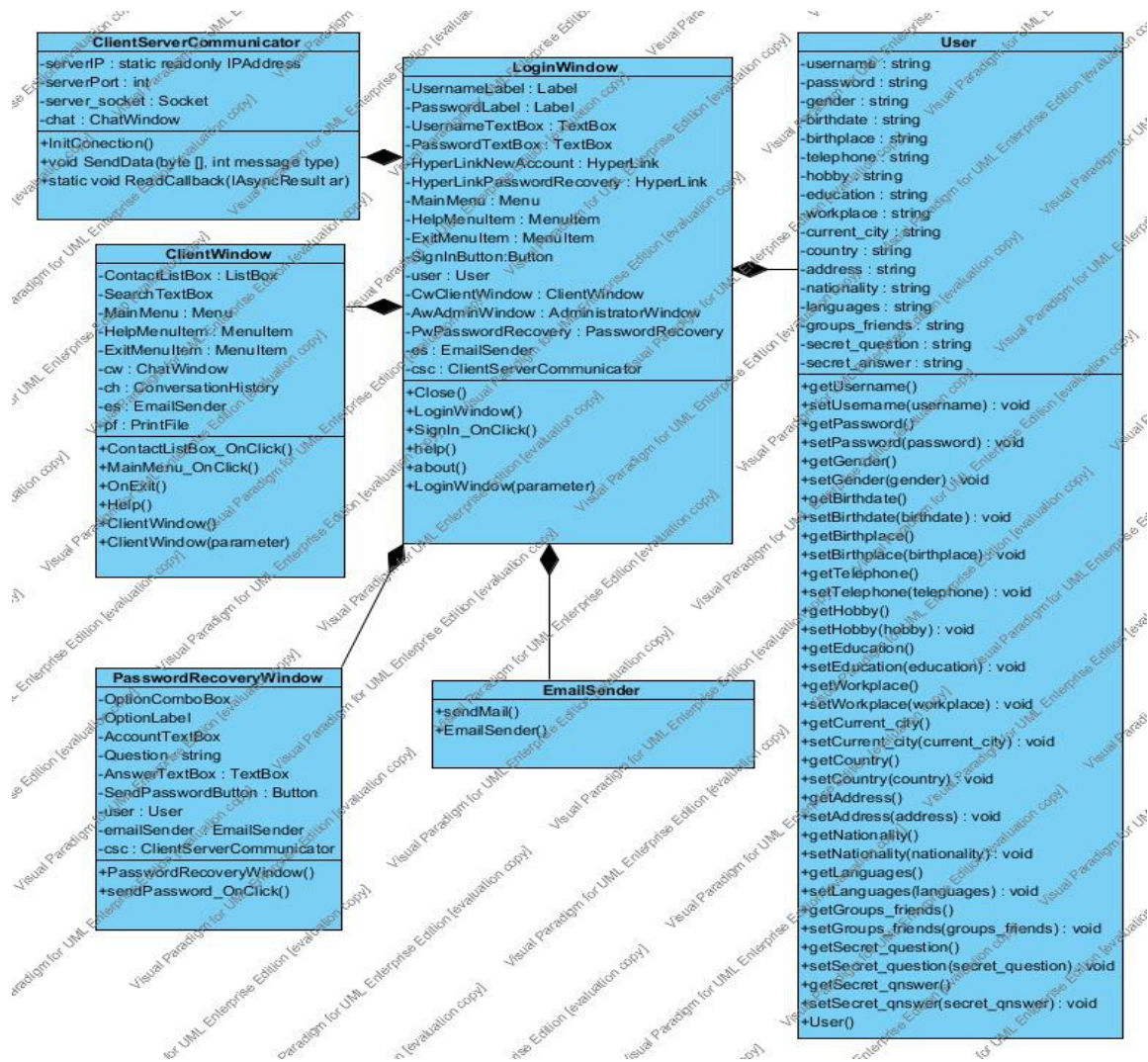
-Initializare conexiunii si setare unei functii pentru care se ocupa cu acceptare conexiunilor

- static void AcceptCallback(IAsyncResult ar)  
-functie care va accepta noile conexiuni si va delega alta functie sa trateze primirea mesajului
- static void ReadCallback(IAsyncResult ar)  
-fiecare mesaj va contine in primele doua campuri dimensiunea mesajului si tipul mesajul, iar aceasta functie va astepta pana cand mesajul a fost primit complet si apoi va fi deserializat si va fi apelata functie de tratare a mesajelor
- void message(StateStatus mess)  
-Anunta toti clientii online de schimbarea statusului
- void message(FriendOp mess)  
-In functie de tipul operatiei trimite efectua modificarile necesare in baza de date
- void message( Message mess, Socket dest)  
-Trimite mai departe mesajul clientului caruia ii este adresat mesajul
- void message( Login mess)  
-Se verifica informatiile de autentificare cu cele din baza de date si in cazul care acestea valideaza utilizatorului i se va permite conexiunea, altfel se va inchide
- void message (Autentificare mess)  
-Se vor adauga noi intrari in baza de date cu privirile la detaliile utilizatorului in cazul in care id este unic si adresa de email respecta formatul. In caz negativ se trimite un mesaj clientului prin care se va cere sa incerce din nou indicand si campul ce trebuie corectat.

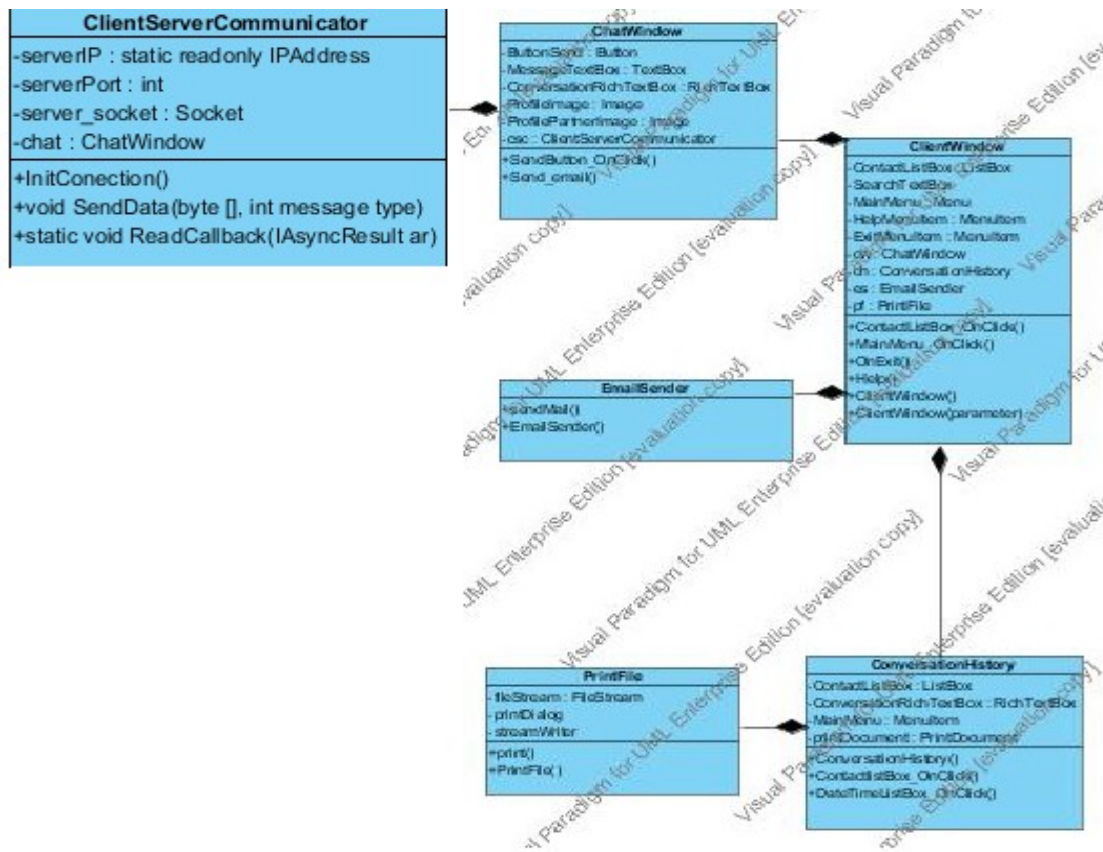
Server
-port : int -socket_server : Socket -baze_date : DataBaseManager -map : Dictionary<name, Socket>
+getPort() +setPort(port) : void +Server(DataBaseManager baze_date) +BindConection() +AcceptCallback(IAsyncResult ar) +ReadCallback(IAsyncResult ar) +void message(StateStatus mess) +message(Message mess, Socket dest) +message(Autentificare mess)

### 3.Use case

#### 3.1 Autentificare



### 3.2 Chat



### 3.3 Administrare

